

RRRRRRRRRRRR		MMM		MMM	SSSSSSSSSSSS
RRRRRRRRRRRR		MMM		MMM	SSSSSSSSSSSS
RRRRRRRRRRRR		MMM		MMM	SSSSSSSSSSSS
RRR	RRR	MMMMMM	MMMMMM	SSS	
RRR	RRR	MMMMMM	MMMMMM	SSS	
RRR	RRR	MMMMMM	MMMMMM	SSS	
RRR	RRR	MMM	MMM	SSS	
RRR	RRR	MMM	MMM	SSS	
RRR	RRR	MMM	MMM	SSS	
RRRRRRRRRRRR		MMM		SSSSSSSSSS	
RRRRRRRRRRRR		MMM		SSSSSSSSSS	
RRRRRRRRRRRR		MMM		SSSSSSSSSS	
RRR	RRR	MMM			SSS
RRR	RRR	MMM			SSS
RRR	RRR	MMM			SSS
RRR	RRR	MMM			SSS
RRR	RRR	MMM			SSS
RRR	RRR	MMM			SSS
RRR	RRR	MMM			SSS
RRR	RRR	MMM		SSSSSSSSSSSS	
RRR	RRR	MMM		SSSSSSSSSSSS	
RRR	RRR	MMM		SSSSSSSSSSSS	

_S

Syn

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

NT

PI

```

RRRRRRRR      MM      MM      SSSSSSSS      000000      UU      UU      PPPPPPPP      DDDDDDDD      AAAAAA      TTTTTTTTTT
RRRRRRRR      MM      MM      SSSSSSSS      000000      UU      UU      PPPPPPPP      DDDDDDDD      AAAAAA      TTTTTTTTTT
RR      RR      MMMM      MMMM      SS      00      00      UU      UU      PP      PP      DD      DD      AA      AA      TT
RR      RR      MMMM      MMMM      SS      00      00      UU      UU      PP      PP      DD      DD      AA      AA      TT
RR      RR      MM      MM      SS      00      0000      UU      UU      PP      PP      DD      DD      AA      AA      TT
RR      RR      MM      MM      SS      00      0000      UU      UU      PP      PP      DD      DD      AA      AA      TT
RRRRRRRR      MM      MM      SSSSSS      00      00      UU      UU      PPPPPPPP      DD      DD      AA      AA      TT
RRRRRRRR      MM      MM      SSSSSS      00      00      UU      UU      PPPPPPPP      DD      DD      AA      AA      TT
RR      RR      MM      MM      SS      0000      00      UU      UU      PP      DD      DD      AAAAAAAAAA      TT
RR      RR      MM      MM      SS      0000      00      UU      UU      PP      DD      DD      AAAAAAAAAA      TT
RR      RR      MM      MM      SS      00      00      UU      UU      PP      DD      DD      AA      AA      TT
RR      RR      MM      MM      SS      00      00      UU      UU      PP      DD      DD      AA      AA      TT
RR      RR      MM      MM      SSSSSSSS      000000      UUUUUUUUUU      DDDDDDDD      AA      AA      TT
RR      RR      MM      MM      SSSSSSSS      000000      UUUUUUUUUU      DDDDDDDD      AA      AA      TT
                                     ....
                                     ....
                                     ....
                                     ....

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

(2) 74
(3) 97

DECLARATIONS
RMSSUPDATE - COMMON \$UPDATE SETUP AND DISPATCH ROUTINE

```
0000 1      $BEGIN RMSOUPDAT,000,RMSRMS,<DISPATCH FOR UPDATE OPERATION>
0000 2
0000 3
0000 4 *****
0000 5 *
0000 6 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 *  ALL RIGHTS RESERVED.
0000 9 *
0000 10 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 *  TRANSFERRED.
0000 16 *
0000 17 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 *  CORPORATION.
0000 20 *
0000 21 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 *
0000 24 *
0000 25 *****
0000 26
0000 27 ++
0000 28 Facility: rms32
0000 29
0000 30 Abstract:
0000 31      this routine is the highest level control
0000 32      routine to perform the $update function.
0000 33
0000 34
0000 35
0000 36 Environment:
0000 37      star processor running starlet exec.
0000 38
0000 39 Author: L F LAVERDURE,      Creation Date: 3-FEB-1977
0000 40
0000 41 Modified By:
0000 42
0000 43 V03-007 JWT0141      Jim Teague      11-Nov-1983
0000 44      Change IFBSV_RUM to IFBSV_ONLY_RU
0000 45
0000 46 V03-006 KPL0003      Peter Lieberwirth      26-Jul-1983
0000 47      If AT jnling, tell RJR this is an UPDATE.
0000 48
0000 49 V03-005 KPL0002      Peter Lieberwirth      24-Jul-1983
0000 50      If AT journaling, get RAB data that describes user's request.
0000 51
0000 52 V03-004 KPL0001      Peter Lieberwirth      20-Jun-1983
0000 53      Change some references to JNLFLG to JNLFLG2.
0000 54
0000 55 V03-003 TMK0001      Todd M. Katz      27-Dec-1982
0000 56      Do not turn off the IRBSV_FIND_LAST flag within RMSRSET. This
0000 57      flag will be turned off within the routines for the individual
```


0000	58	:
0000	59	:
0000	60	:
0000	61	:
0000	62	:
0000	63	:
0000	64	:
0000	65	:
0000	66	:
0000	67	:
0000	68	:
0000	69	:
0000	70	--
0000	71	:
0000	72	:

file organizations.

V03-002 JWH0153 Jeffrey W. Horn 8-Dec-1982
Don't allow \$UPDATE if not in recovery unit and RU only
specified for file.

V03-001 KBT0196 Keith B. Thompson 23-Aug-1982
Reorganize psects

V02-004 REFORMAT D M WALP 25-JUL-1980

RM
Sy

\$S
\$S
\$S
\$S
\$S
CH
DI
DI
ER
EX
FT
IF
IM
IR
IR
IR
IR
IR
PI
RA
RM
RM
RM
RM
RM
SE
SU
SY
SY
TP

PS

RM
SAPh
--
In
Co
Pa
Sy

```
0000 74      .SBTTL  DECLARATIONS
0000 75
0000 76 :
0000 77 : Include Files:
0000 78 :
0000 79 :
0000 80 :
0000 81 : Macros:
0000 82 :
0000 83
0000 84      $IFBDEF
0000 85      $RMSDEF
0000 86      $RJRDEF
0000 87
0000 88 :
0000 89 : Equated Symbols:
0000 90 :
0000 91 :
0000 92 :
0000 93 : Own Storage:
0000 94 :
0000 95
```

```
0000 97 .SBTTL RMS$UPDATE - COMMON $UPDATE SETUP AND DISPATCH ROUTINE
0000 98
0000 99 :++
0000 100 : RMS$UPDATE
0000 101
0000 102 : RMS$UPDATE - this routine performs common rab function setup followed
0000 103 : by dispatch to organization-dependent $update code
0000 104
0000 105 : Calling sequence:
0000 106 :
0000 107 : entered from exec as a result of user's calling sys$update
0000 108 : (e.g., by using the $update macro)
0000 109
0000 110 : Input Parameters:
0000 111 :
0000 112 : ap user's argument list addr
0000 113
0000 114 : Implicit Inputs:
0000 115 :
0000 116 : the contents of the rab and related irab and ifab.
0000 117
0000 118 : Output Parameters:
0000 119 :
0000 120 : r1 destroyed
0000 121 : r0 status code
0000 122
0000 123 : Implicit Outputs:
0000 124 :
0000 125 : various fields of the rab are filled in to reflect
0000 126 : the status of the $update operation. (see rms functional
0000 127 : spec for a complete list.)
0000 128
0000 129 : the irab is similarly updated.
0000 130
0000 131 : a completion ast is queued if specified in the user arglist.
0000 132
0000 133 : Completion Codes:
0000 134 :
0000 135 : standard rms (see functional spec for list).
0000 136
0000 137 : Side Effects:
0000 138 :
0000 139 : none
0000 140
0000 141 :--
0000 142 : $ENTRY RMS$UPDATE
0000 143 : $STPT UPDATE
0000 144 : $RABSET FAC=IFB$V_UPD ; do common setup
0000 145
0000 146 :
0000 147 : returns to user on error
0000 148 :
0000 149
0000 150 : BBC #IFB$V_ONLY RU,IFB$B_JNLFLG(R10),10$ ; branch if not RU only
0000 151 : BBS #IFB$V_RUP,IFB$B_JNLFLG2(R10),10$ ; branch if in RU
0000 152 : RMSERR NRU
0000 153 : BRW RM$EXRMS
```

OE 00A0 CA 00 E1 000A 150
08 00A2 CA 02 E0 0010 151
FFE2' 31 001B 152
001B 153

```

09 00A0 CA 04 E1 001E 154 10$:
    51 1C D0 001E 155
    00000000'EF 16 001E 156 :
    001E 157 : If AT journaling, get some information from RAB.
    001E 158 :
    001E 159 BBC #IFB$V AT,IFB$B_JNLFLG(R10),20$ ; skip if not AT jnling
    0024 160 MOVL #RJR$_UPDATE,R1 ; this is UPDATE
    0027 161 JSB RMSAT_COM_RAB ; get RAB data into RJR
    002D 162 20$:
    002D 163
    002D 164 :
    002D 165 : dispatch to org-dependent code
    002D 166 :
    002D 167
    002D 168 CASE TYPE=B, SRC=IFB$B_ORGCASE(R10),-
    002D 169 DISPLIST=<RMSUPDATE1,RMSUPDATE2,RMSUPDATE3>
    0038 170 .IF NE $$RMSTEST&$$RMS_TBUGCHK
    0038 171 BRW RMSERRORG
    003B 172 .ENDC
    003B 173 .END

```

00000008
FFC5' 31

RMSOUPDAT
Symbol table

DISPATCH FOR UPDATE OPERATION

F 4

16-SEP-1984 01:34:16
5-SEP-1984 16:25:37

VAX/VMS Macro V04-00
[RMS.SRC]RMSOUPDAT.MAR;1

Page 6
(3)

\$\$PSECT EP	=	00000000		
\$\$RMSTEST	=	0000001A		
\$\$RMS_PBUGCHK	=	00000010		
\$\$RMS_TBUGCHK	=	00000008		
\$\$RMS_UMODE	=	00000004		
IFBSB_JNLFLG	=	000000A0		
IFBSB_JNLFLG2	=	000000A2		
IFBSB_ORGCASE	=	00000023		
IFBSV_AT	=	00000004		
IFBSV_ONLY_RU	=	00000000		
IFBSV_RUP	=	00000002		
IFBSV_UPD	=	00000003		
PIOSA_TRACE	*****	X	01	
RJRS_UPDATE	=	0000001C		
RMSAT_COM_RAB	*****	X	01	
RMSERRORG	*****	X	01	
RMSRXRMS	*****	X	01	
RMSRSET	*****	X	01	
RMSUPDATE1	*****	X	01	
RMSUPDATE2	*****	X	01	
RMSUPDATE3	*****	X	01	
RMSUPDATE	=	FFFFFFFFE RG	01	
RMS\$ NRU	=	000187FC		
TPTSC_UPDATE	*****	X	01	

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes														
. ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
RMSRMS	0000003B (59.)	01 (1.)	PIC	USR	CON	REL	GBL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE				
\$ABSS	00000000 (0.)	02 (2.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:01.18
Command processing	111	00:00:00.73	00:00:04.74
Pass 1	221	00:00:05.45	00:00:16.93
Symbol table sort	0	00:00:00.70	00:00:01.30
Pass 2	45	00:00:01.03	00:00:02.00
Symbol table output	4	00:00:00.04	00:00:00.04
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	414	00:00:08.06	00:00:26.22

The working set limit was 1350 pages.
28944 bytes (57 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 583 non-local and 4 local symbols.
173 source lines were read in Pass 1, producing 13 object records in Pass 2.
18 pages of virtual memory were used to define 17 macros.

! Macro library statistics !

Macro library name

Macros defined

\$255\$DUA28:[RMS.OBJ]RMS.MLB;1
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1
\$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

9
1
3
13

696 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RMSOUPDAT/OBJ=OBJ\$:RMSOUPDAT MSRC\$:RMSOUPDAT/UPDATE=(ENH\$:RMSOUPDAT)+EXECML\$/LIB+LIB\$:RMS/LIB

0331 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

														
														
														
														
														
														
														
														
														
														
														
														
														
														
														